

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method of processing data comprising:
defining an ~~object~~instance of a class, the class supporting with
defined fields to support, each of the fields having a field values, a field
name, and a field data type, wherein memory space for each field value is in-
preallocated memory space when the instance of the class is created, and with-
an
options, each of the options having an option value, an option name, and
an option data type, wherein the options are referenceable data structure which-
supports references to option values without preallocation of memory space for
the full-option values when the instance of the class is created;
accessing a selected field value ~~stored in one of the defined fields and accessing~~
~~an option value not stored in the defined fields in the~~ instance using a first single
program code expression, the first single program code expression comprising an
operator and the field name corresponding to the selected field value, object using
expressions of and accessing a selected option value in the instance using a second
single program code expression, the second single program code expression
comprising the operator and the option name corresponding to the selected option value
~~the same syntactic form; and~~

during compilation, determining whether at least one of the expressions
accesses one of (a) a the selected field value or (b) an the selected option value;

when it is determined that the ~~expression accesses a~~ selected field value is being accessed, compiling the first single program code expression into a first code for accessing the selected field value; and

when it is determined that the ~~expression accesses an~~ selected option value is being accessed, compiling the second single program code expression into a second code for accessing the selected option value.

2. (Currently Amended) A method as claimed in claim 1 wherein ~~the a~~ selected option data structure ~~identifies~~ is associated with change handler code that is executed when ~~an~~ the option value corresponding to the selected option changes.

3. (Currently Amended) A method as claimed in claim 2 wherein change handler code ~~for~~ associated with the selected ~~one~~ option is defined in ~~different a~~ plurality of classes within a class inheritance hierarchy and the change handler code from each of the plurality of classes is executed when the option value corresponding to the selected option changes.

4. (Currently Amended) A method as claimed in claim 1, ~~wherein the option~~ data structure includes a type description of the option value, the method further comprising:

during compilation of an operation on the selected option value, using the corresponding option data type description ~~in the option data structure to process an~~ the operation ~~on the option value~~.

5. (Currently Amended) A method as claimed in claim 1 wherein an selected option data structure is associated with ~~includes~~ a default value, the method further comprising, in a get operation to the selected option in an the instance of the class, if ~~an the option value which applies corresponding to the instance the selected option~~ has been set, getting the set option value and, if ~~no the option value which applies corresponding to the selected option~~ has not been set, getting the associated default value ~~for the class~~.

6. (Currently Amended) A method as claimed in claim 1 wherein defining an instance comprises ~~comprising~~:

defining a first instance of a first class, the first class supporting a first set of with a first options data structure of a first form which supports, in instances of the class, references to option values without preallocation of memory space for the full option values, wherein the first set of options is associated with a first listing data structure of a first form, the method further comprising: [[;]]

defining a second instance of a second class, the second class supporting a second set of options with a second option data structure of a second form which supports, in instances of the second class, references to option values without preallocation of memory space for the full option values when the second instance is created, wherein the second set of options is associated with a second listing data structure of a second form, the second form being different from the first form; and

during compilation, encoding an ~~option~~ operation on the first instance as a method call to ~~an object~~ the first instance of the first class without regard to form of the first or second listing data structure; and

during compilation, encoding an operation on the second instance ~~to an object of the second class~~ without regard to the form of the ~~option~~ first or second listing data structure supported by the class.

7. (Currently Amended) A method as claimed in claim 1 further comprising: notifying ~~objects~~ the instance of the class of a change in ~~an~~ the selected option value through a change handler identified by an option binding object, the option binding object being located by first searching a mapping data structure for any previously-computed mapping from the option name corresponding to the selected option value to the option binding object and, if no mapping was ~~previously computed~~ found, by then computing ~~the~~ a mapping from the option name corresponding to the selected option value to the option binding object and storing the mapping in the mapping data structure.

8. (Currently Amended) A method as claimed in claim 1 wherein the instance of the class is associated with a listing data structure, the listing data structure comprising option data structure comprises a linked list of option items,
each of the option items corresponding to a referenced option comprising an option in the instance that has been referenced,

each of the option items having the options values and the option name
corresponding to one of the referenced options, and
the option items being arranged in a linked list;
wherein the method further comprises, when a first option is referenced in order to set a
first option value for the first option, checking the listing data structure for a first option
item corresponding to the first option;
when the first option item is found, setting the first option value in the first option
item; and
when no first option item is found, creating the first option item, setting the first
option value in the first option item, and storing the first option item with the set first
option value in the listing data structure.

9. (Currently Amended) A method as claimed in claim 1 wherein a nonlocal option value applies to other ~~objects~~ instances of the class in a nonlocal option hierarchy.

10. (Original) A method as claimed in claim 9 wherein the nonlocal option hierarchy is a graphical hierarchy.

11. (Currently Amended) A data processing system for processing including ~~data objects~~ instances of a class, the ~~data objects~~ instances of the class comprising:

defined fields, each of the fields having to support a field values, a field name,
and a field data type, wherein memory space for each field value is in preallocated
memory space when an instance of the class is created; and

an options, each of the options having an option value, an option name, and an
option data type, wherein the options are referenceable data structure which supports
references to option values without preallocation of memory space for the full-option
value when the instance of the class is created; the data processing system further
comprising:

program code wherein the a selected field value stored in one of the defined
fields in the instance is accessed using a first single program code expression, the first
single program code expression comprising an operator and the field name
corresponding to the selected field value, and wherein a selected option value not
stored in the defined fields being accessed are in the instance is accessed in the object
using a second single program code expression, the second single program code
expression comprising the operator and the option name corresponding to the selected
option value with expressions of the same syntactic form; and

a compiler which determines whether at least one of the expressions accesses
one of (a) a the selected field value or (b) an the selected option value, wherein,

when the compiler determines that the expression accesses selected a-
field value is being accessed, the compiler compiles the first single program code
expression into a first code for accessing the selected field value; and

when the compiler determines that the ~~expression accesses an~~ selected option value is being accessed, the compiler compiles the second single program code expression into a second code for accessing the selected option value.

12. (Currently Amended) A system as claimed in claim 11 wherein the ~~option data structure a selected option is associated with~~ identifies change handler code that is executed when ~~an~~ the option value corresponding to the selected option changes.

13. (Currently Amended) A system as claimed in claim 12 wherein change handler code associated with the selected ~~for one option~~ is defined in ~~different a~~ plurality of classes within a class inheritance hierarchy and the change handler code from each of the plurality of classes is executed when the option value corresponding to the selected option changes.

14. (Currently Amended) A system as claimed in claim 11 ~~wherein the option data structure includes a type description of the option value, the system further comprising a compiler which, during compilation of an operation on the selected option value, uses the corresponding option data type description in the option data structure to process an~~ the operation ~~on the option value~~.

15. (Currently Amended) A system as claimed in claim 11 wherein an selected option data structure is associated with ~~includes a default value which that is~~

obtained when no ~~option-value~~ has been set in an ~~applicable instance object~~ for the selected option.

16. (Currently Amended) A system as claimed in claim 11 comprising a plurality of classes having data structures supporting options, wherein the options supported by each class are associated with listing data structures having of different forms, and wherein the compiler ~~a compiler which encodes an option operation on instances of the classes~~ as a method call to an instance ~~object~~ of one of the classes without regard to the form of the ~~option-listing~~ data structure ~~supported by the class.~~

17. (Currently Amended) A system as claimed in claim 11 further comprising change handlers which notify ~~objects~~ instances of the class of a change in an supported option value, and

a mapping data structure which maps an supported option name and class to an option binding object, wherein the option binding object ~~which~~ identifies a change handler corresponding to the option.

18. (Currently Amended) A system as claimed in claim 11,
wherein the instance of the class is associated with option a listing data structure ~~comprises comprising a linked list of option items having option values,~~
each of the option items corresponding to a referenced option comprising
an option that has been referenced in the instance,

each of the option items having the option value and the option name
corresponding to one of the referenced options, and
the option items being arranged in a linked list;
wherein, when a first option is referenced in order to set a first option value for
the first option, the linked list is checked for a first option item corresponding to the first
option,
when the first option item is found, the first option value is set in the first
option item; and
when no first option item corresponding to the first option is found, the first
option item is created, the first option value is set in the first option item, and the
first option item with the set first option value is stored in the listing data structure.

19. (Currently Amended) A system as claimed in claim 11 wherein a nonlocal option value applies to other ~~objects~~ instances of the class in a nonlocal option hierarchy.

20. (Original) A system as claimed in claim 19 wherein the nonlocal option hierarchy is a graphical hierarchy.

21. (Currently Amended) A data processing system comprising:
means for defining an ~~object~~ instance of a class, the class supporting with-

defined fields to support, each of the fields having a field values, a field name, and a field data type, wherein memory space for each field value is in-
preallocated when the instance of the class is created, and

~~memory space and with an options,~~ each of the options having an option value, an option name, and an option data type, wherein the options are
referenceable data structure which supports references to option values without
preallocation of memory space for the full-option values when the instance of the
class is created;

means for accessing a selected field value stored in one of the defined fields and
~~accessing an option value not stored in the defined fields in the~~ instance using a first
single program code expression, the first single program code expression comprising an
operator and the field name corresponding to the selected field value, object using
expressions and accessing a selected option value in the instance using a second
single program code expression, the second single program code expression
comprising the operator and the option name corresponding to the selected option value
of the same syntactic form; and

means for determining, during compilation, whether at least one of the
expressions accesses one of (a) a the selected field value or (b) ~~an~~ the selected option
value;

when it is determined that the ~~expression accesses a~~ selected field value
is being accessed, compiling the first single program code expression into a first
code for accessing the selected field value; and

when it is determined that the ~~expression accesses an~~ selected option value is being accessed, compiling the second single program code expression into a second code for accessing the selected option value.

22. (Currently Amended) A computer program product comprising:
- a computer usable medium for storing data; and
- a set of computer program instructions embodied on the computer usable medium, including instructions ~~for to~~:
- ~~define an object~~ defining an instance of a class, the class supporting with
- defined fields, each of the fields having a field to support values, a field
- name, and a field data type, wherein memory space for each field value is in-
- preallocated memory space when the instance of the class is created and with an
- options, each of the options having an option value, an option name, and
- an option data type, wherein the options are referenceable data structure which
- supports references to option values without preallocation of memory space for
- the full-option values when the instance is created;
- accessing a selected field value ~~stored in one of the defined fields and access an~~
- ~~option value not stored in the defined fields in the~~ instance using a first single program
- code expression, the first single program code expression comprising an operator and
- the field name corresponding to the selected option value, and accessing a selected
- option value in the instance using a second single program code expression, the
- second single program code expression comprising the operator and the option name

corresponding to the selected option value object using ~~expressions of the same~~
syntactic form; and

during compilation, ~~determine~~ determining whether at least one of the
expressions accesses one of (a) a the selected field value or (b) an the selected option
value;

when it is determined that the ~~expression accesses a~~ selected field value
is being accessed, compiling the first single program code expression into a first
code for accessing the selected field value; and

when it is determined that the ~~expression accesses an~~ selected option
value is being accessed, compiling the second single program code expression
into a second code for accessing the selected option value.

23. (Currently Amended) A product as claimed in claim 22 wherein the
computer program instructions include instructions to notify ~~objects~~ instances of the
class of a change in an supported option value.

24. (Currently Amended) A product as claimed in claim 22,
wherein the options are associated with data structure comprises a linked
list of option items,
each of the option items corresponding to an option that has been referenced in
the instance, and
each of the option items having the option values and the option name
corresponding to one of the referenced options; and
wherein the linked list is checked when a first option is referenced in order to set
a first option value for the first option, and
when the first option item is found, the first option value is set in the first option
item; and
when no first option item corresponding to the first option is found, the first option
item corresponding to the first option is created, the first option value is set in the first
option item, and the first option item with the set first option value is stored in the linked
list.

25-30. (Cancelled).